Advanced Access Control System for Mobility as a Service Platform

Anjali Rajith
Service Systems Innovation Center
Research and Development Group, Hitachi Ltd.
Yokohama, Japan
email: anjali.rajith.he@hitachi.com

Soki Sakurai
Service Systems Innovation Center
Research and Development Group, Hitachi Ltd.
Yokohama, Japan
email: soki.sakurai.mk@hitachi.com

***Abstract*—Mobility as a Service (MaaS) is a digital platform that integrates multiple transport and third-party providers into a single channel that enables customers to easily book and pay for services. Data misuse, accidental data leakage, and malicious services are the key threats to confidential customer and service provider data. To eliminate these threats, an advanced access control system is proposed for MaaS that utilizes a context based, customer-centric, hybrid fine-grained and quantitative trust computing approach. The eXtensible Access Control Language architecture is extended by adding a trust score computation module and policy update function. When a data request arrives, the data access context is determined, and the trust score of the data requester is computed based on selected trust parameters. Access to confidential information is subjected to trust score condition and fine-grained policy rules. Real-time policy updating and data masking are performed when personal data-sharing preferences are changed. Our model ensures a safe, reliable data flow and mitigates the security and privacy issues.**

Keywords-MaaS; access control; trust score; privacy; XACML.

I. INTRODUCTION

Mobility as a Service (MaaS) [1]–[3] is the integration of multiple transport providers and service providers into a single digital platform, accessible on demand. The integration and unification are undertaken by intermediaries who are between supply side and demand side as shown in Figure 1. The supply side is made up of Mobility Service Providers (MSPs), public or private organizations that own and manage transport services through transport service providers and other mobility-related services. The demand side is the customers or end-users who avail the MaaS service. A stack of services in the middle layer are required to coordinate the users and services of MaaS platform, such as payment services, ticketing services, recommendation services, etc. Therefore, the data in the MaaS platform belong to a plethora of services and customers.

In this work, MaaS architecture is proposed to be built on top of Lumada [4], the IoT platform of Hitachi, Figure 2. Lumada provides a stack of basic and solution functions, such as artificial intelligence, security, etc. required to build a business solution. It comprises: a Data Zone that captures and collects IT data from web applications and databases, and OT data from IoT data, such as weather data, information from Road Service Units (RSUs), and GPS data; Data Flow that acquires data from various sources; Data Processing Governance that governs the data processing, and Data understanding layer that provides several tools to understand the data. However,

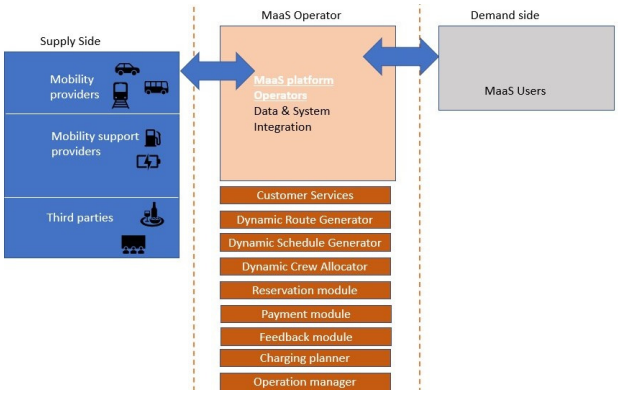

Figure. 1. Basic concept of MaaS.

the entire MaaS architecture is outside the scope of this work and only the Data Processing Governance Layer is considered that is related to technologies that handle the data management, policy handling, and access control. Stringent laws and regulations govern personal information, making data provenance and access monitoring imperative for keeping track of unauthorized access attempts on the MaaS platform. The Service Level Agreement (SLA) and access control policies in this layer help to prevent the invocation of services by unauthorized operators and prevent malicious services from accessing sensitive information.

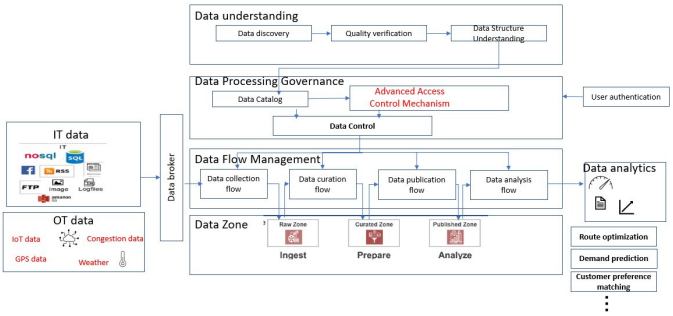

Figure. 2. Lumada-based Maas architecture.

A. MaaS Security Challenges

A large amount of Personally Identifiable Information (PII) and travel-related information of users and drivers are shared via the platform. Accidental or purposeful leakage of such information can breach users' privacy. MaaS also faces various realistic threats in which attackers gain access to GPS position-

 ISBN: 978-1-68558-007-0

ing of vehicles, steal data, and thereby gives undue advantages to competitors. There are also insider threats, such as misusing information for purposefully favouring or destroying one's business. Thus, we focus on three main security challenges faced by the MaaS platform [5]:

1) Misuse of personal information of customers: Sensitive personal information of the customers is utilized by third-parties without their knowledge. Power of consent doesn't always fall in the hands of the customers. Therefore, the first challenge is to ensure that the system complies with the personal information protection acts, as well as provides the power of consent revocation and right to determine who access their data to the customers.
2) Accidental data leakage: The service providers participating in the MaaS platform receive way too much information than that is actually required. So, the second challenge is to ensure that only minimum necessary information is provided to the services.
3) Threats from malicious services: There are chances that malicious insider services utilize the data for favouring their own business. The third challenge is to continuously monitor trust factor of the participating parties in a quantitative manner based on agreed SLAs.

B. Our Contributions

Our main contributions are as follows:

1) We propose a novel context based, customer-centric, hybrid fine-grained and quantitative trust computing access control approach such that the access approval to a particular data resource is determined based on the dynamic context of data access, access policy rules, and trust score of the data requester calculated at the time of access based on historic access log parameters.
2) A system architecture design with additional functionalities is proposed by extending the standard access control architecture to address the security challenges.

This paper is organized as follows: Section II explains related work on MaaS security threats and various access control systems. Section III explains conventional access control architecture and our system's approach. Section IV explains our system's architecture and details each component. Section V explains the implementation. Section VI concludes the paper and mentions future works on optimizing the system.

II. Related Work

There are different works that attempt to study the security threats faced by MaaS platforms. Callegati et. al [5], [6] focused on the insider threats in MaaS and followed a tiered architecture from individual operators to markets of federated MaaS providers to classify the threats of each tier, and proposed appropriate mitigation measures. These works point out the necessity of proper access control technologies and security loopholes caused due to inadequate policy definitions and indicate the necessity of adequate access logging and auditing facilities.

Some works address the security and privacy issues through access control approaches in cloud environments and blockchains. Toahchoodee et al. [7] proposed a trust-based access control approach for access control of pervasive computing systems, whereas P. K. Behera and P. M. Khilar [8] proposed a trust-based access control approach for cloud environment, in which the user request is passed through various sub-modules to make the authorization decision. However, this work is related to access of cloud resources where user should submit the QoS requirements, such as security, cost, computing power, etc., and the user authorization is made solely based on user trust value computed by a trust management module. A. Singh and K. Chatterjee [9] proposed a mutual trust based access control model for the healthcare system by modifying the conventional access control system by integrating the trust degree of communicating parties in the access control system. However, this system does not account for any dynamic changes in the data requesters and the environment. The access decision that is solely based on trust score of few parameters could make the system vulnerable to attacks that track the access decision pattern. Trust and reputation systems are widely used in e-commerce, social networks, search engines, and so on. User scoring is performed based on their activities in scoring systems of credit card agencies [10]. Works such as, A. Josang [11] and G. Zacharia and P.Maes [12] proposed trust and reputation systems in online environment by storing records of activities of users and calculating reputation score for users.

Works, such as Hogan et al. [13] used blockchain in MaaS for improving the transactional aspects and increasing the trust between various actors involved in MaaS. Guo et al. [14] also studied blockchain-based access control. However, blockchain is not considered in this work considering the high computation and gas cost. Ammar et al. [15] has implemented a semantic handler component for deciding the context of data access. In our work, we only consider two contexts, and hence do not implement a separate module for context evaluation.

None of the works have explicitly studied the access control paradigm for a cross-industrial collaboration system, such as MaaS, where the data providers and the data requesters change in a dynamic manner. The security preferences and data exchange between various operators, services, and end-users can change in an dynamic environment and that makes it very challenging.

III. Approach

This section explains the standard XACML architecture in sub-section III-A and proposed architecture in sub-section III-B.

A. Standard XACML Architecture

Access Control systems are trust infrastructures that allow or restrict access to protected resources through data authorization and access control. Role-Based Access Control (RBAC) [16] [17] system assigns access permissions to roles, and roles to subject. However, RBAC implements a static permission

 ISBN: 978-1-68558-007-0

list and cannot scale into real-world dynamic environments. Attribute-Based Access Control (ABAC) [18]–[20] defines an attribute-based access control paradigm in which access rights are granted to users through eXtensible Access Control Language (XACML) [21]–[23] policies that combines the subject, object, action, and environment attributes. This approach is dynamic to an extent that access decision is based on attribute values at the time of access attempt. It comes with an architecture with the components, Policy Enforcement Point (PEP), Policy Decision Point (PDP), Policy Administration Point (PAP), and Policy Information Point (PIP). In the standard XACML engine, PEP wraps the data request into a XACML request and communicates it to the PDP. The PDP with the help of PIP, checks the attribute values and verifies the policies managed by PAP, makes an access decision, and communicates it back to PEP.

The fine-grained ABAC approach actualizes a dynamic access control technique, where only necessary information is passed on to the data requesters based on complex policies and rules. However, based on the studies in section II, implementing a conventional standalone access control system is insufficient for tackling the security challenges of a highly dynamic transaction environment, such as MaaS. Therefore, we devised a context-based hybrid access control approach that adds new functional modules to the standard access control system. In addition to pre-defined access rules and conditions stipulated through fine-grained policies, a separate module is necessary to modify and update conditions in the policy rules in a real-time manner. The conventional access control approach neither supports attribute logging nor quantitative computation. Logging the access attribute values and access decisions related to all data requesters helps to evaluate their reliability. The trust score computation approach calculates the trust score at the time of data request, based on the historic log data. It is to be noted that access log information is collected on a mutual consensus with participating parties. The trust score computation approach is critical in issuing security warnings to the admin user and the data requesters, restricting access to confidential customer information, and auditing. Since the system collects subjective trust parameters, such as user feedback, it can be used as a means to provide service provider recommendations to like-minded customers. Therefore, we implement new modules on top of the XACML architecture to incorporate additional functionalities.

B. Overview of Proposed System

MaaS is a highly dynamic data transaction environment involving multiple data providers and data requesters, with lot of security issues which are pointed out in sections I and II. In this paper, we propose a context based, customer-centric, hybrid access control approach for minimizing security issues faced by the MaaS platform. Our hybrid approach combines the fine-grained access control mechanism and quantitative trust computing approach on top of the XACML architecture. Trust score of the data requesters are computed at the time of data access request, based on selected trust parameters. The hybrid approach incorporates the trust score threshold condition into the XACML policies such that access to sensitive information could mandate to satisfy trust score criteria in the policy rules. In the MaaS platform, it is important to give more power to the user to make decision on the access of his/ her personal information. The customer-centric approach dynamically updates the access control policies based on real-time customer personal data access security preferences and dynamic trust threshold value changes. The context-based approach considers two contexts, normal and emergency; that is evaluated by the context handler in the gateway, based on access attributes. The data control flow is slightly altered in the emergency context, such that a risk threshold attribute is set and access control to sensitive customer information is passed to authorities for emergency evacuations. A data masking or data transformation function is incorporated such that even during the unfortunate event of accidental data leakage, sensitive information is protected.

IV. Architecture Design

Figure 3 represents the overall architecture and control flow of the proposed advanced access control system.

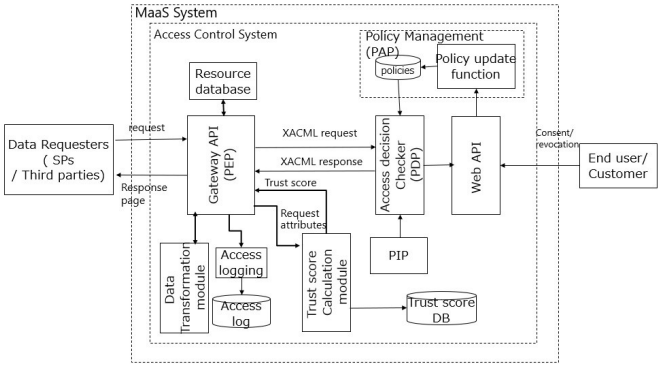

Figure. 3. System architecture to realize the proposed approach.

The main components are as follows:

1) Gateway API - Receives the data access request/ response and has a context evaluator function.
2) Web API - A user interface that allows customers/ end users to interact with the MaaS data management layer.
3) Policy Management (PAP) - Manages the policies and triggers the real-time policy update function upon changes.
4) Access Logging - Logs the access parameters of the data requesters for trust score calculation.
5) Trust calculation module - Calculates trust score of the data requesters and stores in trust score database.
6) Data Transformation Module - Masks the sensitive customer information.
7) Access decision checker (PDP) - Provides the access decision based on the policies defined.

In this work, WSO2 Identity Server [24], an open source software is used for access control, policy management, and configuration of PIP.

 ISBN: 978-1-68558-007-0

When an access request for a data resource arrives, the first step is to verify the credentials, which is performed by an identity authentication module. The details of this module will be skipped as it does not fall under the scope of this work. In the second step, data request attributes are extracted by the gateway API, the PEP. In standard XACML architecture, the PEP module generates the authorization request and sends it to the PDP module immediately. In our approach, the context evaluator function in the gateway API judges the data request context, which is a dynamic attribute with values `[normal, emergency]`. If the context value is `normal`, the request attributes are passed on to a trust calculation module. If the context value is `emergency`, step three is skipped. In step three, the trust calculation module performs the trust score calculation of the data requester based on historic access log data and communicates the trust score back.

As the fourth step, the PEP generates the XACML request based on the attributes and context. In the fifth step, the authorization request is passed on to the access decision checker, that checks the access request parameters against defined policies. Based on the access decision, access is permitted or denied to the data resource requested. In the sixth step, either the requested data fetched from resource database or an error message is delivered by the gateway to the data requester. The access request parameters and response parameters are logged accordingly for auditing and trust scoring purpose.

Definition 1. *The XACML data flow is as follows:*
Step 1: The gateway API receives the access request and the context value is determined based on $[subject_{real-time}, environment_{real-time}, action_{real-time}.]$
Step 2: If context value is `emergency`*, the request handler generates the XACML request with value of* `risk threshold` *attribute set as 1, and skip to step 4, else go to step 3.*
Step 3: The trust calculation module calculates the trust score of the data requester and sends back to gateway.
Step 4: The gateway generates the XACML request based on the values of access attributes, risk threshold, and trust score and communicates the request to access decision checker (PDP).
Step 5: The access decision checker evaluates the access request by checking additional attributes against PIP and access policies, provides response to gateway, which is either `permit` *or* `deny`*.*
Step 6: If the response is `permit`*, gateway executes obligation services and provides access to requested data to the data requester. Else, an authorization error message is returned. The response and access parameters are logged to access log database.*

A. Context Evaluation

In the context-based approach, data access context is determined by the context evaluator function in the gateway API module. In normal scenarios, the access control system prevents access to any unauthorized access attempts on sensitive end-user information, such as user location and sensitive driver information, such as driver GPS location, name, etc. Only users in the appropriate role can access this information. This is to prevent the insider misuse of information that is completely irrelevant to their purpose. However, in the emergency life-threatening situations, such as natural disasters, the location information of all users can be accessed by the admin user or city authorities for necessary actions. For this purpose, the dynamic context attribute is utilized. Based on the values of $[subject_{real-time}, action_{real-time}, environment_{real-time}]$ attributes, the context is determined as `[normal, emergency]`. The data flow handling is altered in the `emergency` context by skipping the trust computation based on agreed SLAs with end-users.

B. Customer-Centric Approach

In the customer-centric approach, the customers are given the power to control access to their personal information. This is realized through the policy management and the web API module. They can enable or disable access to their PII, as well as location and destination information to selected transport providers and third-party services by setting their security preferences through the web API.

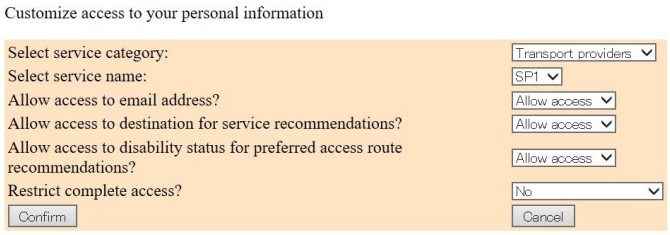

Figure. 4. User security preference form.

Figure 4 illustrates the user security preference form available to the MaaS users. It provides the option to restrict complete record access or column access to selected service categories and service providers. If an end-user allows/ restricts his information to be accessed by a third-party service through the provided web API, the access rules associated with the records of user in the respective location are updated real-time by calling an `update function`, that can update the attribute values in policy conditions and add or delete conditions to the policy rules. A XACML policy template is auto-generated with the new attribute values or conditions, and the corresponding customer policy is updated by invoking the policy administration APIs of WSO2 [25] tool. Once the policy is updated and deployed, the change in access permissions to concerned service providers are reflected.

Figure 5 illustrates the restricted column access, in which the data requester cannot access the columns, such as *disability status* of the customer. The values of selected columns by the

 ISBN: 978-1-68558-007-0

customer are masked by calling a `mask function` offered by data transformation module.

Customer Name	Email address	Destination	Disability status	Smoking preference
sam	sam1@gmail.com	bokutho hosptial	N	not smoking
cathy	******@gmail.com	****	****	smoking
yamamoto	yamamoto99@gmail.com	bakurocho station	N	not smoking
suzuki	******@yahoo.com	****	****	not smoking
miura	******@hotmail.com	****	****	not smoking
akiko	akiko@gmail.com	sumida community center	N	not smoking

Figure. 5. Transformed data.

C. Access Logging

The access logging function logs the data request and response attributes associated to a data requester, such as authentication id, name, service category, access location, access time, action, resource id, and access decision to an access log database, which are used for the trust score computation and auditing purpose. The parameters associated with trust computation, which will be explained later in sub-section IV-D, are derived values from the log data. Any unfortunate incident of access by an unauthorized person is reported and prompts the admin for immediate policy definitions' review. This is very important because careless policy definitions can breach SLAs with customers and service providers.

D. Trust Scoring Approach

This section explains the trust scoring approach, parameters used, and trust score calculation. The trust scoring function is realized through the trust computation module. It calculates the trust score of the data requesters based on the trust parameters logged in the access log database, as well as from security monitoring system. The trust parameters under consideration are:

1) Invalid data access request rate: This parameter is calculated for the data requester based on the unauthorized access attempts obtained from the access history data logs. The invalid data access attempt rate, DAR is calculated as:
$$DAR = \frac{R_d}{R_t} \tag{1}$$
where, R_d is the number of unauthorized or failed access attempts made by the data requester and R_t is the total number of access attempts.
2) Access frequency rate: The ratio of number of access requests from a particular data requester to total number of access requests per unit time. Monitoring this parameter helps to detect Denial-of-Service(DOS) Attacks. Access frequency rate, AFR is calculated as:
$$AFR = \frac{R_t}{T_t} \tag{2}$$
where, R_t is the number of access attempts made by the data requester and T_t is the total number of access attempts per unit time.
3) Transaction rate: The number of successful transactions made by a service provider through the MaaS platform with respect to other service providers who belong to the same user category per unit time interval.
4) User satisfaction: This is a subjective parameter based on the user feedback about the service of particular service provider. This could be considering aspects, such as punctuality in the service, delay notifications, payment service, etc.
5) Network Protection: A weighted impact score calculated by security monitoring system of MaaS system on each collaborating service provider, based on data on network-related parameters, such as access measures (encryption measures), network environment (local or remote), and suspicious packet count.

The trust score of the data requester *n*, T^n is calculated as:
$$T^n = \sum_{w=i}^{j} w_i * p_i \tag{3}$$
where p_i are the *j* parameters used, and w_i are the weights of p_i parameters. The weights vary from 1 to 10 based on priority. Higher weights are assigned to high priority parameters, based on occurrence of security incidents. All new users are assigned a minimum trust score greater than 0. The calculated trust scores are stored in a trust database. The trust score, T^n is normalized such that T_{max}, T_{min} are the minimum and maximum trust scores and the direction of reliability is made similar:
$$NT^n = \frac{T^n - T_{min}}{T_{max} - T_{min}} \tag{4}$$
An access control criteria can be set such that such that trust score of the data requester calculated at time of access must be greater than defined score $NT^n >= NT^{th}$, where NT^{th} is the threshold value. NT^{th}, T_{max}, and T_{min} are selected based on the training dataset. The threshold is selected such that it maximizes the accuracy of trust decision. Some malicious users may attempt to take advantage of the system before their trust score drops. To handle this scenario, the weights associated with negative trust parameters are changed based on the user activity. Weights of positive parameters such as, transaction rate and user satisfaction are unchanged irrespective of user activity.

E. Access Decision Checker

The access decision checker is the PDP entitlement engine. Here, WSO2 identity server is used. The XACML request generated from the gateway contains the attribute values associated with the data requester. The access decision is made by PDP by checking the request against the attribute values obtained from PIP and policies. If response from PDP is `permit`, access is allowed. All PDP responses are logged and a `permit` decision to the data requester triggers an email to the admin user through XACML-obligation features. The hybrid approach formulates an access decision based on fine grained policy-based conditions such that the trust score

 ISBN: 978-1-68558-007-0

based condition can be incorporated to policies in a flexible or optional manner.

V. Implementation

In this section, the implementation of proposed model in an experimental MaaS system is explained. To implement the system, Eclipse Integrated Development Environment (IDE) using Java programming language is used. Capturing of access parameters of the data requesters, customer feedback, and personal information preferences are also done in the same environment. As explained in section IV, WSO2 Identity Server is used as access decision engine. Gateway API (PEP) is implemented as Java servlets running on top of Tomcat server [26].

We have considered ten customers and four service providers (data requesters) in the test set. A dynamic trust threshold monitoring function updates the trust threshold variable in the policy, upon change in optimal threshold value based on past 'n' unit time. The WSO2 policy administration APIs are invoked to update the trust threshold value defined in the policies based on the output of this function. Meeting the trust threshold criteria can be used as a condition in policy rules to access customer records. This criteria can be applied to any confidential records. The customer preferences on their personal record access and trust threshold criteria are reflected in policy rules as shown below:

Definition 2. *rule Rule1* {
description: "Only service providers of category transport providers, with trust score greater than 0.6 are allowed by customer1 to read the data"
subject:"SP1"
action:read
object:"customer#1.data"
condition:"SP1.service_category==transport_provider && $SP1.trust\_score \geq 0.6$"
decision: permit}

Data access will be denied to the requesters who fail to satisfy the policy conditions. Figure 6 illustrates an example XACML policy where decision of the rule is permit, if trust score is greater than the set trust threshold value 0.6. The real-time trust score computation along with the other fine-grained policy rules based on attribute values at the access time provides a better approach when compared to standard models. Since the access decision does not solely depend on trust computation, it can be also be optionally removed as a policy criteria and can be only used for security monitoring purpose.

Figure 7 demonstrates the trust score of the service providers belonging to a selected category service using open-source Grafana dashboard [27].

VI. Conclusion and Future Work

We implemented a context based, customer-centric, hybrid access control system for a MaaS platform to mitigate the

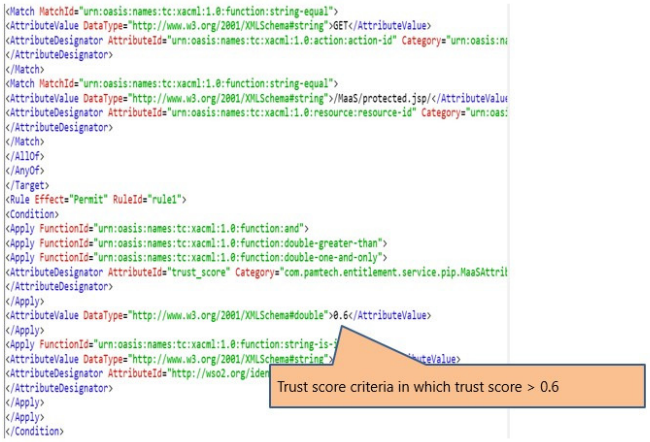

Figure. 6. XACML policy with real-time trust score condition.

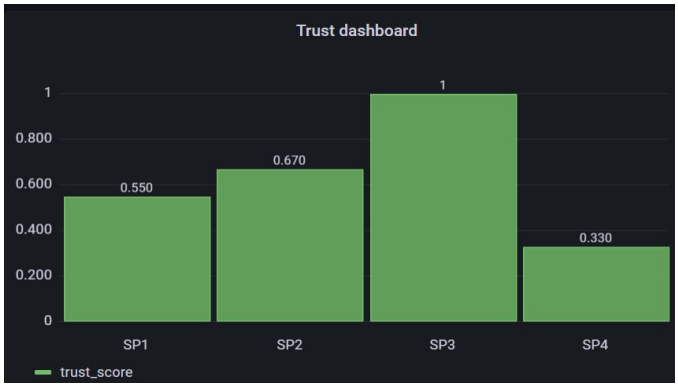

Figure. 7. Trust scores of service providers.

MaaS security challenges, such as misuse of customer information, accidental leakage of sensitive information, and insider threats from malicious services. Our proposed system extends the XACML architecture to address problems caused by malicious users and services. We analyze historic access logs and security monitoring data to derive trust score of the data requesters. Confidential information access is only permitted if the trust score calculated at the time of access and other attribute values meet the stipulated policy rules. Therefore, the access decision made using the hybrid access control model is more reliable than the conventional models. The user-centric approach of this system gives complete power to the end-users in deciding how their personal data is utilized. Data masking and access policy updating are done real-time without affecting other processes in the system. The context-based approach classifies the data access into `normal` and `emergency` contexts. This module prioritizes safety over security by altering the data flow handling in the emergency context. The advanced access control system can be realized in any dynamic data collaboration platform similar to MaaS.

In future work, we will study the dynamic selection of trust parameters based on historical parameter data analysis. We will also study the system performance by analysing the number of concurrent users against the number of cores with

 ISBN: 978-1-68558-007-0

respect to response time and computation cost. Furthermore, experiments will be performed for the empirical analysis of trust score to study rate of change of trust score, ideal trust score retention period, and effect of few suspicious transactions on a trust-worthy user.

ACKNOWLEDGMENT

This research is funded and supported by Hitachi Research and Development Group, and would like to thank them for providing support and technical advice.

REFERENCES

[1] "Maas https://smart-maas.eu/en/ (visited on 05/20/2021)."
[2] D. Sitányiová and S. Masarovicová, "Development status of sustainable urban mobility plans in european union new member states," *International Journal of Transport Development and Integration*, vol. 1, no. 1, pp. 16–27, 2016.
[3] E.-I. Europe, "Mobility as a service (maas) and sustainable urban mobility planning," *ERTICO–ITS Europe: Brussels, Belgium*, 2019.
[4] S. Hanaoka, Y. Taguchi, T. Nakamura, H. Kato, T. Kaji, H. Komi, N. Moriwaki, N. Kohinata, K. Wood, T. Hashimoto, *et al.*, "Iot platform that expands the social innovation business," *Hitachi Review*, vol. 65, no. 9, p. 439, 2016.
[5] F. Callegati, S. Giallorenzo, A. Melis, and M. Prandini, "Cloud-of-things meets mobility-as-a-service: An insider threat perspective," *Computers and Security*, vol. 74, 11 2017.
[6] F. Callegati, S. Giallorenzo, A. Melis, and M. Prandini, "Data security issues in maas-enabling platforms," in *2016 IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI)*, pp. 1–5, 2016.
[7] M. Toahchoodee, R. Abdunabi, I. Ray, and I. Ray, "A trust-based access control model for pervasive computing applications," in *IFIP Annual Conference on Data and Applications Security and Privacy*, pp. 307–314, Springer, 2009.
[8] P. K. Behera and P. M. Khilar, "A novel trust based access control model for cloud environment," in *Proceedings of the international conference on signal, networks, computing, and systems*, pp. 285–295, Springer, 2017.
[9] A. Singh and K. Chatterjee, "A mutual trust based access control framework for securing electronic healthcare system," in *2017 14th IEEE India Council International Conference (INDICON)*, pp. 1–6, IEEE, 2017.
[10] "Ficoscore https://www.fico.com/ (visited on 11/02/2021)."
[11] A. Jøsang, "Trust and reputation systems," in *Foundations of security analysis and design IV*, pp. 209–245, Springer, 2007.
[12] G. Zacharia and P. Maes, "Trust management through reputation mechanisms," *Applied Artificial Intelligence*, vol. 14, no. 9, pp. 881–907, 2000.
[13] G. Hogan, S. Dolins, I. Senturk, I. Fyrogenis, Q. Fu, E. Murati, F. Costantini, and N. Thomopoulos, "Can a blockchain-based maas create business value?," vol. 28, p. 1, 10 2019.
[14] H. Guo, W. Li, M. Nejad, and C.-C. Shen, "Access control for electronic health records with hybrid blockchain-edge architecture," in *2019 IEEE International Conference on Blockchain (Blockchain)*, pp. 44–51, IEEE, 2019.
[15] N. Ammar, Z. Malik, E. Bertino, and A. Rezgui, "Xacml policy evaluation with dynamic context handling," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 9, pp. 2575–2588, 2015.
[16] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-based access control models," *Computer*, vol. 29, no. 2, pp. 38–47, 1996.
[17] E. Coyne and T. R. Weil, "Abac and rbac: scalable, flexible, and auditable access management," *IT professional*, vol. 15, no. 03, pp. 14–16, 2013.
[18] E. Yuan and J. Tong, "Attributed based access control (abac) for web services," in *IEEE International Conference on Web Services (ICWS'05)*, IEEE, 2005.
[19] R. S. Sandhu, "Attribute-based access control models and beyond.," in *AsiaCCS*, p. 677, 2015.
[20] S. Quirolgico, V. Hu, T. Karygiannis, *et al.*, *Access Control for SAR Systems*. US Department of Commerce, National Institute of Standards and Technology, 2011.
[21] "Xacml https://en.wikipedia.org/wiki/XACML (visited on 08/02/2021)."
[22] Y. Keleta, J. Eloff, and H. Venter, "Proposing a secure xacml architecture ensuring privacy and trust," *Research in Progress Paper, University of Pretoria*, 2005.
[23] C. D. P. K. Ramli, H. R. Nielson, and F. Nielson, "The logic of xacml," *Science of Computer Programming*, vol. 83, pp. 80–105, 2014.
[24] "Wso2 identity server https://github.com/wso2/product-is (visited on 06/04/2021)."
[25] "Wso2 policy administration apis https://is.docs.wso2.com/en/latest/develop/entitlement-with-apis/ (visited on 06/04/2021)."
[26] "Apache tomcat server https://tomcat.apache.org/ (visited on 12/09/2020)."
[27] "Grafana https://grafana.com/ (visited on 07/08/2021)."

 ISBN: 978-1-68558-007-0